PHN 16762

ITEM 1

Europäisches Patentamt | European Patent Office | Office européen des brevets

Bescheinigung | Certificate | Attestation

Die angehefteten Unterlagen stimmen mit der ursprünglich eingereichten Fassung der auf dem nächsten Blatt bezeichneten europäischen Patentanmeldung überein.

The attached documents are exact copies of the European patent application described on the following page, as originally filed.

Les documents fixés à cette attestation sont conformes à la version initialement déposée de la demande de brevet européen spécifiée à la page suivante.

Patentanmeldung Nr. | Patent application No. | Demande de brevet n°

97402763.3

Der Präsident des Europäischen Patentamts;
Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets
p.o.

M.W. Graham

Europäisches Patentamt
European Patent Office
Office européen des brevets

Blatt 2 der Bescheinigung
Sheet 2 of the certificate
Page 2 de l'attestation

Anmeldung Nr.:
Application no.: 97402763.3
Demande n°:

Anmeldetag:
Date of filing: 17/11/97
Date de dépôt:

Anmelder:
Applicant(s):
Demandeur(s):
Philips Electronics N.V.
5621 BA Eindhoven
NETHERLANDS

Bezeichnung der Erfindung:
Title of the invention:
Titre de l'invention:
A method which inherently achieves the same advantages of advanced prediction mode in H.263 encoders

In Anspruch genommene Priorität(en) / Priority(ies) claimed / Priorité(s) revendiquée(s)

Staat: State: Pays:	Tag: Date: Date:	Aktenzeichen: File no. Numéro de dépôt:

Internationale Patentklassifikation:
International Patent classification:
Classification internationale des brevets:
H04N7/26

Am Anmeldetag benannte Vertragstaaten:
Contracting states designated at date of filing: AT/BE/CH/DE/DK/ES/FI/FR/GB/GR/IE/IT/LI/LU/MC/NL/PT/SE
Etats contractants désignés lors du depôt:

Bemerkungen:
Remarks:
Remarques:

A method which inherently achieves the same advantages of Advanced Prediction Mode in H.263 encoders

Daniele Bagni† Gerard de Haan ‡

†Philips S.p.A, Philips Research Monza
Via Philips, 12, 20052 Monza (MI), Italy
Tel: +39-39-203 7804 Fax: +39-39-203 7800
E-mail: bagni@monza.research.philips.com

‡Philips Research Laboratories,
Television Systems Group,
Prof. Holstlaan 4, (WO01)
5656 AA Eindhoven, The Netherlands.
Tel: +31-40-742555 Fax: +31-40-742630
E-mail: dehaan@natlab.research.philips.com

ABSTRACT

This paper presents a method to obtain temporally predicted pictures with a quality that actually is achievable only when H.263 video encoders can apply the negotiable option called Advanced Prediction Mode. Furthermore we expect that our method will probably decrease both the bitrate and the computational complexity. Our proposal aims to achieve the same advantages of Advanced Prediction Mode, even when this last one is not used in H.263 low bitrate video encoding. The solution is fully H.263 standard compatible but not yet standardized. It can be used at CIF, QCIF and SQCIF resolution.

1 INTRODUCTION

The H.263 standard for low bitrate video–conferencing [1]–[2] is based on a video compression procedure which exploits the high degree of spatial and temporal correlation in natural video sequences. The hybrid DPCM/DCT coding removes temporal redundancy using inter–frame motion compensation. The residual error images are further processed by block Discrete Cosine Transform (DCT), which reduces spatial redundancy by de–correlating the pixels within a block and concentrating the energy of the block itself into a few low order coefficients. The DCT coefficients are then quantized according to a fixed quantization matrix that is scaled by a Scalar Quantization factor (SQ). Finally, Variable Length Coding (VLC) achieves high encoding efficiency and produces a bitstream, which

is transmitted over ISDN (digital) or PSTN (analogue) channels, at constant bitrates. Due to the intrinsic structure of H.263, the final bitstream is produced at variable bitrate, hence it has to be transformed to constant bitrate by the insertion of an output buffer which acts as feedback controller. The buffer controller has to achieve a target bitrate with consistent visual quality, low delay and low complexity. It monitors the amount of bits produced and dynamically adjusts the quantization parameters, according to its fullness status and to the image complexity.

For videoconferencing on ISDN lines, H.263 can achieve low bitrates in the range of 64–256 kbps with picture formats such as CIF (352 pixels per 288 lines) and QCIF (176 pixels per 144 lines), depending on the scene complexity. Resulting from the compatibility with the first historically videoconferencing standard, H.261 [3], bitrates greater than 256 kbps are also possible. The maximum bitrate for videophone systems on PSTN lines is 20 kbps, achievable only with QCIF and SQCIF (128 pixels per 96 lines) format pictures and actual modems limited at 28.8 kbps.

The H.263 coding standard defines the techniques to be used and the syntax of the bitstream. There are some degrees of freedom in the design of the encoder. The standard puts no constraints about important processing stages such as motion estimation, adaptive scalar quantization, and bit–rate control.

In this paper we introduce a new method for H.263 low bitrate video encoders and decoders, to achieve a high quality temporally predicted pictures. Similar quality is actually achievable only in H.263 terminals that make use of the so called Advanced Prediction Mode, one of the four H.263 negotiable options. The method is a proper motion vectors post–filtering (MVPF), to be applied between the motion estimation and motion compensation in the encoding terminal, or before the motion compensation in the decoding terminal. Even if it should be quite independent on the motion estimation strategy, we will present it jointly to our new motion estimator, since we think that the best performances will be achieved through the joint action of the MVPF with the new motion estimator. The main advantages of our proposal are the following: 1) no perceptible degradation of the final image quality, when looking at real time videoconferencing sequences, 2) increase of the transmission channel capacity, 3) decrease of the computational complexity of the motion estimation stage. The proposal can be used at CIF, QCIF and SQCIF resolution.

The organisation of this paper is as follows: Section 2 and 3 summarise the main features of H.263 standard and Advanced Prediction Mode, respectively. In Section 4 and 5 we introduce the new motion estimation and the motion vectors post–filtering. Section 6 presents some experiments with the H.263 standard. Finally, some conclusions are drawn in Section 7 and the invention claims are reported on Section 8.

2 OVERVIEW OF H.263 STANDARD

As shown in Fig. 1, the H.263 video compression is based on an inter-frame DPCM/DCT encoding loop: there is a motion compensated prediction from a previous image to the current one and the prediction error is DCT encoded. At least one frame is a reference frame, encoded without temporal prediction. Hence the basic H.263 standard has two types of pictures: I-pictures that are strictly intra-frame encoded and P-pictures that are temporally predicted from earlier frames.

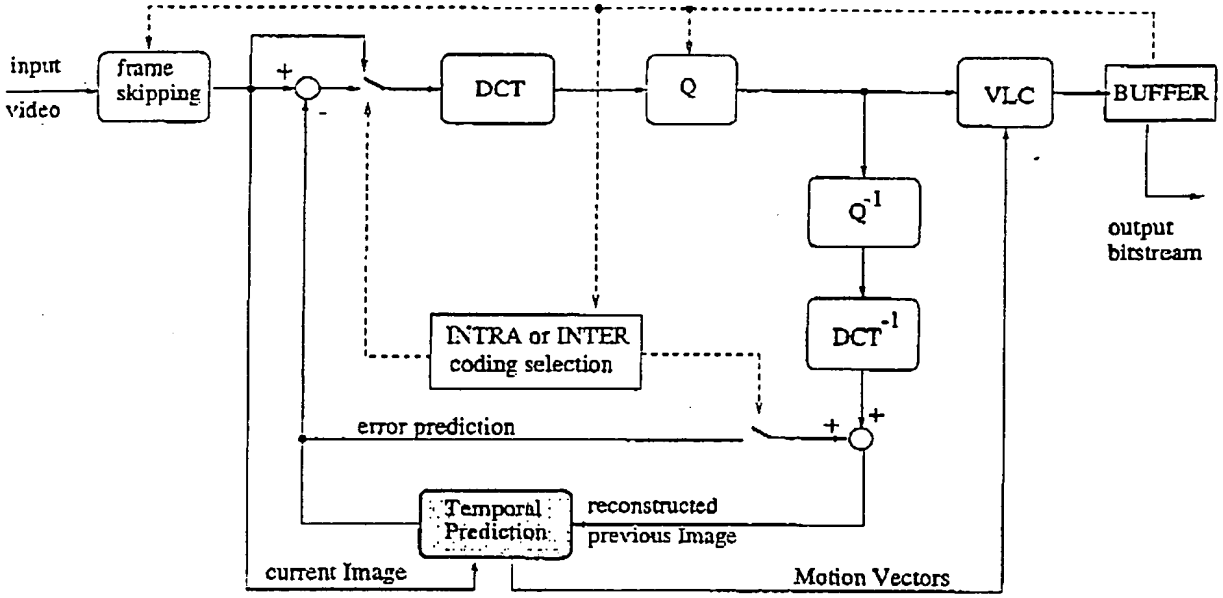

Figure 1: Basic DPCM/DCT video compression block diagram.

The H.263 standard defines a hierarchical bit-stream syntax. There are four hierarchy layers: Picture level, Group Of Blocks level (GOB), macroblock level (MB) and block level. This last is the elementary unit over which DCT operates, it consists of $8 \cdot 8$ pixels. A macroblock is composed by four luminance (Y) blocks, covering a $16 \cdot 16$ area in a picture, and two chrominance blocks (U and V), due to the lower chrominance resolution (see Fig. 2).

The basic H.263 motion estimation and compensation stages operate on macroblocks. The coarseness of quantization is defined by a quantization parameter for the first three layers and a fixed quantization matrix which sets the relative coarseness of quantization for each coefficient. Frame skipping is also used as a necessary way to reduce the bitrate while keeping an acceptable picture quality. As the number of skipped frames is normally variable and depends on the output buffer fullness, the buffer regulation should be related in some way to frame skipping and quantizer step size variations.

A set of Four Negotiable Options can further improve the performance of H.263 in comparison to H.261. They are named respectively Advanced Prediction Mode (APM), Unrestricted Motion Vectors (UMV), Syntax based Arithmetic Coding (SAC) and PB-Frames (PBF). The first option, APM, performs motion estimation and temporal prediction on

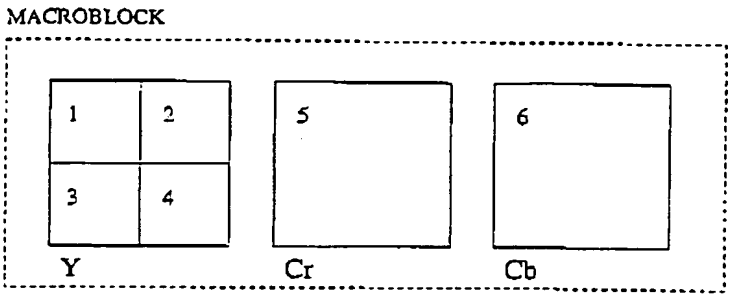

Figure 2: The MB structure.

four $8 \cdot 8$ blocks instead of only one $16 \cdot 16$ macroblock. The second option, UMV, reduces artefacts on the image boundaries since the motion vectors are allowed to point outside the coded picture area. The third option, SAC, can reduce the final bit-rate better then traditional Huffman VLC, but it needs more computational power. Finally, PB-Frames are bi-directionally predicted coded pictures always belonging to a pair of frames; this coding constraint allows to exploit bi-directional temporal prediction without introducing relevant overhead. Therefore, the complete H.263 standard provides three frame coding modes: I, P and PB. Further details about H.263 standard can be found in [1]-[2].

2.1 Motion estimation

In the H.263 main profile, one motion vector per MB is assigned, The motion estimation strategy is not specified, but the motion vectors range is fixed to $[-16, +15.5]$ pixels in a picture for both components. This range can be extended to $[-31.5, +31.5]$ only when the UMV and APM options are jointly used. Every macroblock vector (MV) is then differential encoded with a proper VLC. The prediction vector (PMV), to be added to the vector differences, is obtained as the median value of the three surrounding MB vectors ($MV1, MV2, MV3$), according to:

$$PMV = median(MV1, MV2, MV3)$$

Hence, every PMV component is the median value of the three candidate predictors for this component (see case A of Fig. 3).

In the special cases at the borders of the current GOB or picture the following decision rules are applied in increasing order:

- the candidate predictor $MV1$ is set to zero if the corresponding macroblock is outside the picture (at the left side, as in case B of Fig. 3);
- the candidate predictors $MV2$ and $MV3$ are set to $MV1$ if the corresponding macroblocks are outside the picture (at the top) or outside the GOB (at the top, as in case D of Fig. 3);
- the candidate predictor $MV3$ is set to zero if the corresponding macroblock is outside the picture (at the right side, as in case C of Fig. 3);

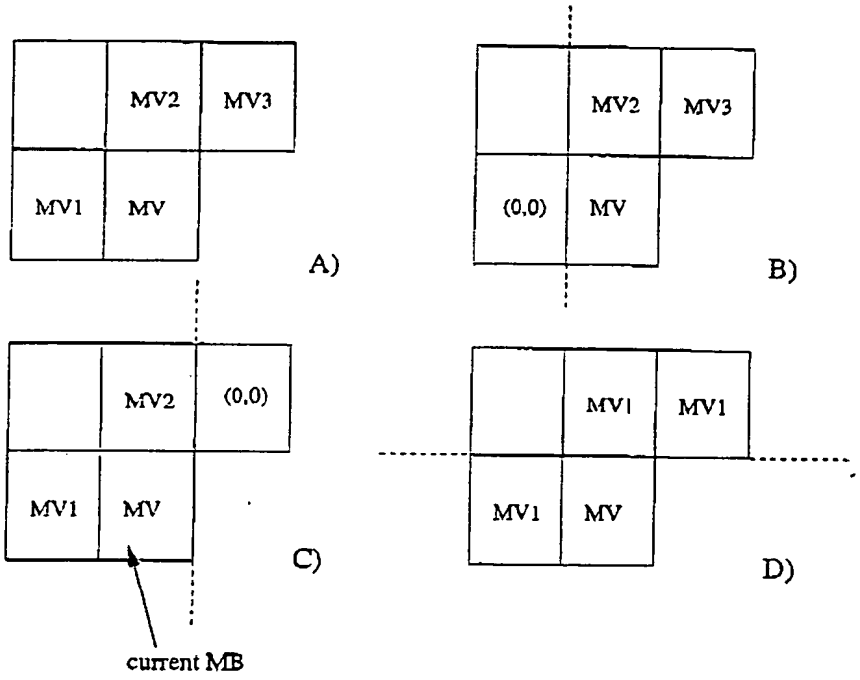

Figure 3: Motion vector prediction for a macroblock.

- when the corresponding macroblock was coded in INTRA mode (if not in PB-Frames mode) or was not coded at all, the candidate predictor is set to zero.

The motion estimation plays a fundamental role in the encoding process, since the quality of temporally predicted pictures strongly depends on the motion vectors accuracy and reliability. The temporal prediction block diagram is shown in Fig. 4.

3 THE ADVANCED PREDICTION MODE

This Section describes the optional Advanced Prediction mode of H.263. Such option includes overlapped block motion compensation and the possibility of four motion vectors per macroblock. The capability of this mode is signalled by external means (for example Recommendation H.245 [5]).

In APM, motion vectors are allowed to cross picture boundaries as is the case in UMV option. The extended motion vector range feature of UMV is not automatically included in the APM, and only it is active if the UMV option is selected. If APM is used in combination with the PBF option, overlapped motion compensation is only used for prediction of the P–pictures, not for the B–pictures.

In APM, one motion vector per MB can be assigned, if the temporal error prediction is lower when four block vectors are used (which are then encoded and transmitted) instead of only one macroblock vector (which is then encoded and transmitted), according to a

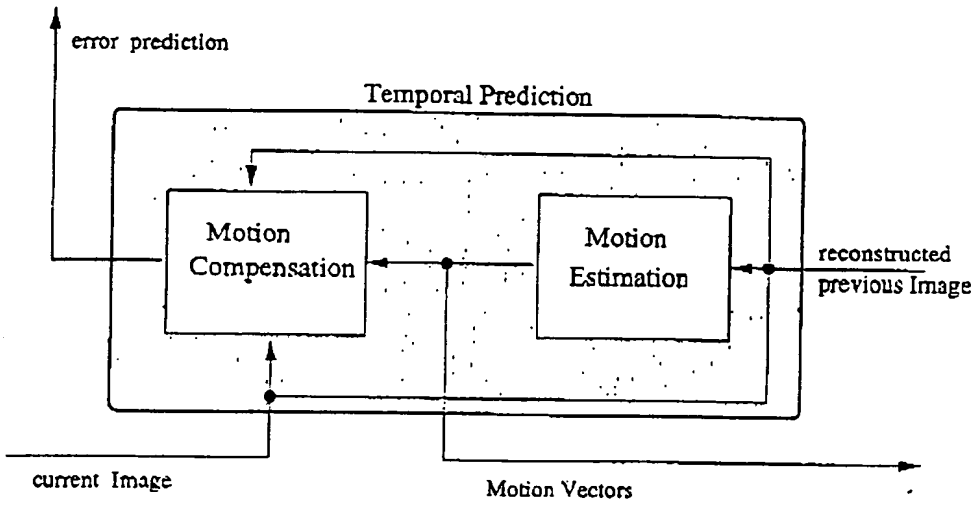

Figure 4: Temporal prediction block diagram.

proper criterion. The one or four vectors decision is indicated by the "MCBPC" codeword for each macroblock. If only one motion vector is transmitted for a certain macroblock, this is defined as four vectors with the same value. If MCBPC indicates that four motion vectors are transmitted for the current macroblock, the information for the first motion vector is transmitted as the codeword "MVD" and the information for the three additional motion vectors is transmitted as the codewords "MVD2–4".

The vectors are obtained by adding predictors to the vector differences indicated by MVD and MVD2–4 in a similar way as when only one motion vector per macroblock is present. Again the predictors are calculated, by median filtering, separately for the horizontal and vertical components. However, the candidate predictors $MV1$, $MV2$ and $MV3$ are redefined as indicated in Fig. 5.

The numbering of the motion vectors is equivalent to the numbering of the four luminance blocks as given in Fig. 2. Motion vector for both chrominance blocks is derived by calculating the sum of the four luminance vectors and dividing this sum by 8; the component values of the resulting sixteenth pixel resolution vectors are modified towards the nearest half pixel position

3.1 Overlapped motion compensation

The APM specifies also the luminance and chrominance motion compensation technique. Each luminance pixel in a block is a weighted average of three prediction values. In order to obtain the three prediction values, two motion vectors are used besides of the motion vector of the current luminance block (MVc): the motion vector of the block at the left or right side of the current luminance block (respectively MVl or MVr), and the motion vector of the block above or below the current luminance block (respectively MVa or MVb). Remote motion vectors from other GOBs are used in the same way as remote motion vectors inside the current GOB.

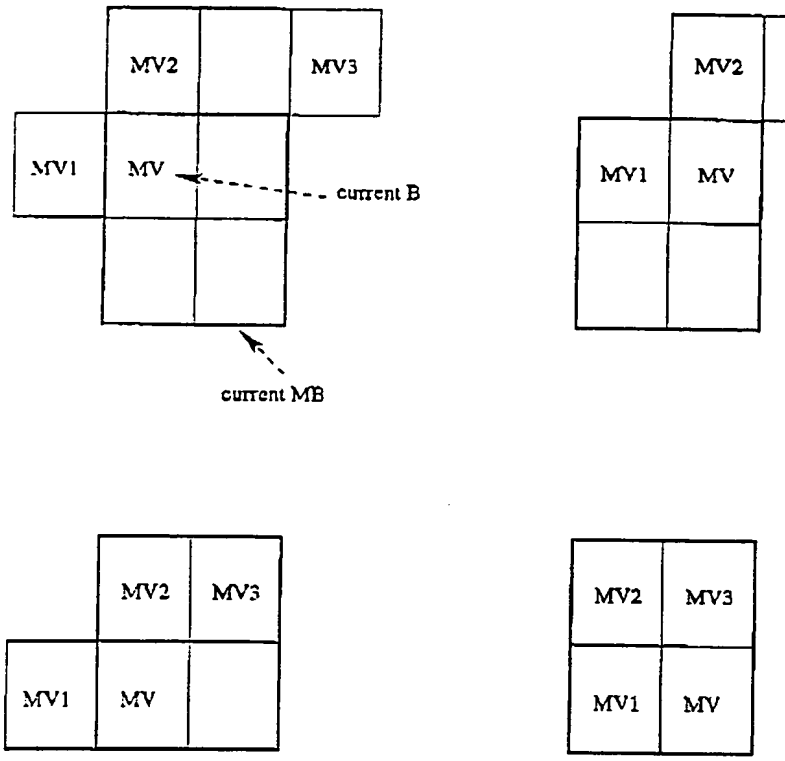

Figure 5: Redefinition of the candidate predictors MV1, MV2 and MV3 for each of the luminance blocks in a macroblock.

For each pixel, the remote motion vectors of the blocks at the two nearest block borders are used. This means that for the upper half of the block the motion vector corresponding to the block above the current block is used, while for the lower half of the block the motion vector corresponding to the block below the current block is used. Similarly, for the left half of the block the motion vector corresponding to the block at the left side of the current block is used, while for the right half of the block the motion vector corresponding to the block at the right side of the current block is used, as shown in Fig. 6.

In our proposal we still continue to use overlapped block motion compensation as it is specified in the H.263 APM. Our invention is focused only on the motion estimation part of APM.

4 THE NEW MOTION ESTIMATION

For estimating the true–motion from a sequence of pictures we departed from the high quality 3–Dimensional Recursive Search block matching algorithm, presented in [7] and [8]. **Unlike the more expensive full–search block matchers that estimate all the possible displacements within a search area, this algorithm only investigates**

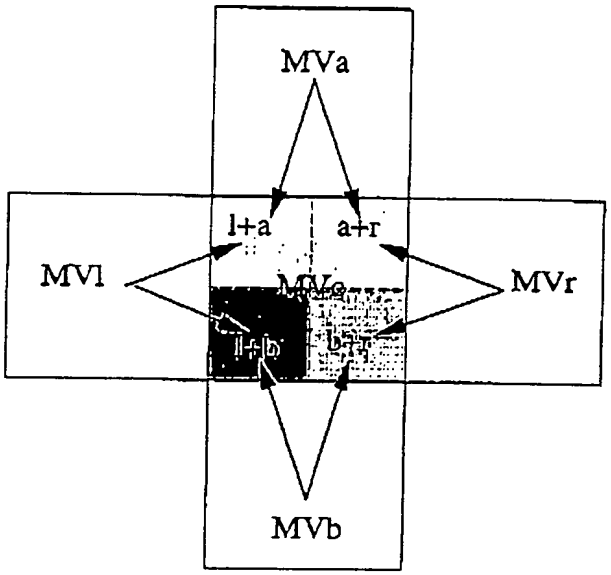

Figure 6: Motion vectors used in the overlapped block motion compensation.

a very limited number of possible displacements. By carefully choosing the candidate vectors, a high performance can be achieved, approaching almost true motion, with a low complexity design. Its attractiveness was earlier proven in an IC for SD–TV consumer applications [10].

4.1 Basic concepts

In block–matching motion estimation algorithms, a displacement vector, or motion vector $\vec{d}(\vec{b}_c, t)$, is assigned to the centre $\vec{b}_c = (x_c, y_c)^{tr}$ of a pixels block $B(\vec{b}_c)$ in the current image $I(\vec{x}, t)$, where tr means transpose. The assignment is done if $B(\vec{b}_c)$ matches a similar block within a search area $SA(\vec{b}_c)$, also centred at $\vec{b}_c$, but in the previous image $I(\vec{x}, t-T)$, with $T = nT_q$ (n integer) representing the time interval between two subsequent decoded images. The similar block has a centre which is shifted with respect to $\vec{b}_c$ over the motion vector $\vec{d}(\vec{b}_c, t)$. To find $\vec{d}(\vec{b}_c, t)$, a number of candidate vectors $\vec{C}$ are evaluated applying an error measure $e(\vec{C}, \vec{b}_c, t)$ to quantify block similarity. Figure 7 illustrates the procedure.

The pixels in the block $B(\vec{b}_c)$ have the following positions:

$$(x_c - X/2 \leq x \leq x_c + X/2)$$
$$(y_c - Y/2 \leq y \leq y_c + Y/2)$$

with X and Y the block width and block height respectively, and $\vec{x} = (x, y)^{tr}$ the spatial

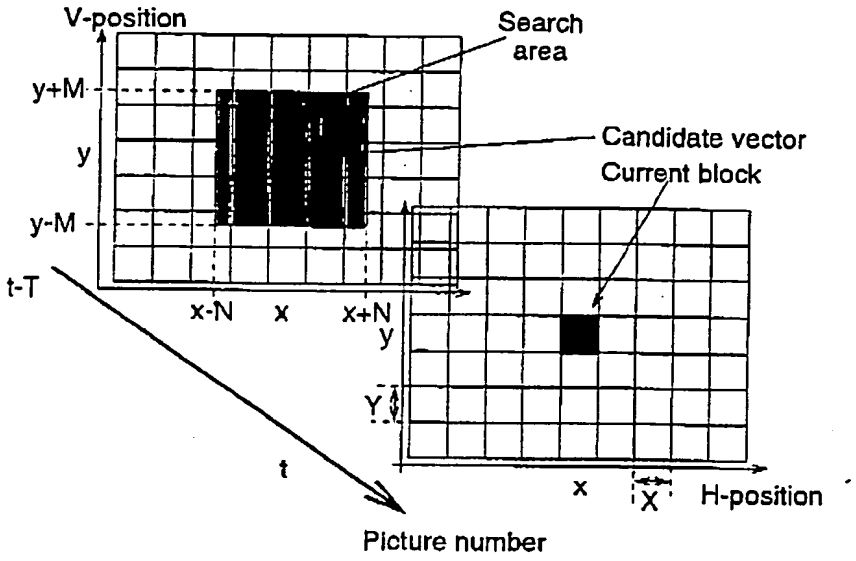

Figure 7: Illustration of block–matching.

position in the image.

The candidate vectors are selected from the candidate set $CS(\vec{b}_c, t)$, which is determined by:

$$CS(\vec{b}_c, t) = \begin{Bmatrix} (\vec{d}(\vec{b}_c - \begin{pmatrix} X \\ Y \end{pmatrix}, t) + \vec{U}_1(\vec{b}_c), \\ (\vec{d}(\vec{b}_c - \begin{pmatrix} -X \\ Y \end{pmatrix}, t) + \vec{U}_2(\vec{b}_c), \\ (\vec{d}(\vec{b}_c - \begin{pmatrix} 0 \\ -2Y \end{pmatrix}, t - T)) \end{Bmatrix} \quad (1)$$

where the update vectors $\vec{U}_1(\vec{b}_c)$ and $\vec{U}_2(\vec{b}_c)$ are randomly selected from an update set US, defined as:

$$US(\vec{b}_c) = US_i(\vec{b}_c) \cup US_f(\vec{b}_c)$$

with the integer updates $US_i(\vec{b}_c)$ stated by:

$$US_i(\vec{b}_c) = \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 0 \\ -2 \end{pmatrix}, \begin{pmatrix} 2 \\ 0 \end{pmatrix}, \begin{pmatrix} -2 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 3 \end{pmatrix}, \begin{pmatrix} 0 \\ -3 \end{pmatrix}, \begin{pmatrix} 3 \\ 0 \end{pmatrix}, \begin{pmatrix} -3 \\ 0 \end{pmatrix} \right\} \quad (2)$$

The fractional updates $US_f(\vec{b}_c)$, necessary to realise half–pixel accuracy, are defined by:

$$US_f(\vec{b}_c) = \left\{ \begin{pmatrix} 0 \\ \frac{1}{2} \end{pmatrix}, \begin{pmatrix} 0 \\ -\frac{1}{2} \end{pmatrix}, \begin{pmatrix} \frac{1}{2} \\ 0 \end{pmatrix}, \begin{pmatrix} -\frac{1}{2} \\ 0 \end{pmatrix} \right\} \quad (3)$$

Either $\vec{U}_1(\vec{b}_c)$ or $\vec{U}_2(\vec{b}_c)$ equals the zero update.

From these Equations it can be concluded that the candidate set consists of spatial and spatio–temporal prediction vectors from a 3–D neighborhood and an updated prediction vector. This implicitly assumes spatial and/or temporal consistency. The updating process involves updates added to either of the spatial predictions. Figure 8 shows where the spatial and spatio–temporal prediction vectors are located relative to the current block.

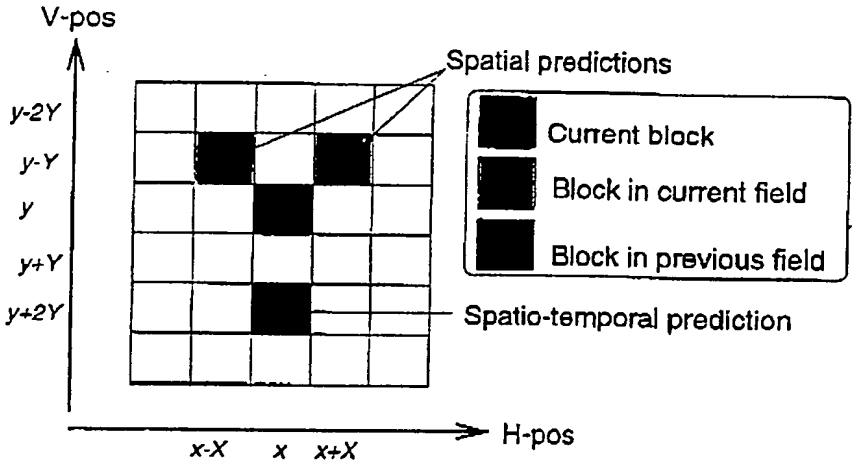

Figure 8: Positions of the prediction vectors relative to the current block.

The displacement vector $\vec{d}(\vec{b}_c, t)$, resulting from the block–matching process, is a candidate vector $\vec{C}$ which yields the minimum value of the error function $e(\vec{C}, \vec{b}_c, t)$:

$$\vec{d}(\vec{b}_c, t) = \{\vec{C} \in CS | e(\vec{C}, \vec{b}_c, t) \leq e(\vec{V}, \vec{b}_c, t)) \quad \forall (\vec{V} \in CS(\vec{b}_c, t))\} \tag{4}$$

The error function is a cost function of the luminance values, $I(\vec{x}, t)$, and those of the shifted block from the previous field, $I(\vec{x} - \vec{C}, t - T)$, summed over the block $B(\vec{b}_c)$. A common choice, which we also use, is the Sum of the Absolute Differences (SAD). The error function is defined by:

$$\begin{aligned} e(\vec{C}, \vec{b}_c, t) &= SAD(\vec{C}, \vec{b}_c, t) \\ &= \sum_{\vec{x} \in B(\vec{b}_c)} |I(\vec{x}, t) - I(\vec{x} - \vec{C}, t - T)| \end{aligned} \tag{5}$$

The vector range is ± 31.5 pixel in both directions, as in the joint application of UMV and APM.

4.2 Iterative estimation

To further improve the motion field consistency, the estimation process is iterated several times, using the motion vectors calculated in the previous iteration to initialize the current iteration, as temporal candidate vectors.

During the first and the third iterations, both previous and current images are scanned from top to bottom and from left to right, that is in the "normal video" scanning direction. On the contrary, the second and fourth iteration are executed with both the images scanned in "anti–video" direction, from bottom to top and from right to left.

The candidate vectors are selected from the new candidate set $CS'(\vec{b}_c, t)$, defined by:

$$CS'(\vec{b}_c; t) = \begin{Bmatrix} \vec{d}(\vec{b}_c - \begin{pmatrix} X \\ (-1)^{i+1}Y \end{pmatrix}, t) + \vec{U}_1(\vec{b}_c), \\ \vec{d}(\vec{b}_c - \begin{pmatrix} -X \\ (-1)^{i+1}Y \end{pmatrix}, t) + \vec{U}_2(\vec{b}_c), \\ \vec{d}_i \end{Bmatrix}$$

where

$$\vec{d}_i = \vec{d}(\vec{b}_c - \begin{pmatrix} 0 \\ -2Y \end{pmatrix}, t - T)$$

for $i = 1$, at every first iteration on an image pair, and

$$\vec{d}_i = \vec{d}(\vec{b}_c - \begin{pmatrix} 0 \\ (-1)^i 2Y \end{pmatrix}, t)$$

for $i \geq 2$, with i indicating the current iteration number.

Furthermore, the first and second iteration are applied on pre–filtered copies of the two decoded images and without sub–pixel accuracy, while the third and fourth iteration are done directly on the original (decoded) images and produce a half–pixel accurate motion vectors.

The pre–filtering consists of a horizontal average over four pixels:

$$I_{pf}(x, y, t) = \frac{1}{4} \sum_{k=1}^{4} I(x \underline{div} 4 + k, y, t) \tag{6}$$

where $I(x, y, t)$ is the luminance value of the current pixel, $I_{pf}(x, y, t)$ is the correspondent filtered version and $\underline{div}$ is the integer division. Two are the main advantages of pre–filtering prior to motion estimation: the first is an increase of the vector field coherency, due to the "noise" reduction effect of the filtering itself, the second is a decrease of the computational complexity, since the sub–pixel accuracy is not necessary in this case.

The computational complexity of the motion estimation is practically independent on the actual (variable) frame rate, for $n \leq 4$. In fact, the number of iterations per images pair varies according to the time interval between two decoded pictures, as shown in Table 1. When $n \geq 5$, we use the same iterations as with $n = 4$.

4.3 Block subsampling, block overlapping and pixel subsampling

It is possible to decrease the computational price of the motion estimation by halving the number of block vectors calculated, that is by using *block subsampling* [7, 8]. The

time interval $T = nT_q$	skipped images	iterations on pre-filtered images	iterations on original (dec.) images
$n = 1$	0	0	0
$n = 2$	1	1	1
$n = 3$	2	1	2
$n = 4$	3	2	2

Table 1: Relation between iterations number and time interval.

subsampled block grid is arranged in a quincunx pattern. If $\vec{d_m} = \vec{d}(\vec{b_c}, t)$ is a missing vector, it can be calculated from the horizontally neighboring available ones $\vec{d_a}$, according to the following formula

$$\vec{d_m} = median(\vec{d_l}, \vec{d_r}, \vec{d_{av}}) \tag{7}$$

where

$$\vec{d_l} = \vec{d_a}(\vec{b_c} - \begin{pmatrix} X \\ 0 \end{pmatrix}, t)$$
$$\vec{d_r} = \vec{d_a}(\vec{b_c} + \begin{pmatrix} X \\ 0 \end{pmatrix}, t)$$
$$\vec{d_{av}} = \frac{1}{2}(\vec{d_t} + \vec{d_b})$$

and

$$\vec{d_t} = \vec{d_a}(\vec{b_c} - \begin{pmatrix} 0 \\ Y \end{pmatrix}, t)$$
$$\vec{d_b} = \vec{d_a}(\vec{b_c} + \begin{pmatrix} 0 \\ Y \end{pmatrix}, t)$$

The median interpolation acts separately on the horizontal and vertical components of the motion vectors. From one iteration to the following we change the subsampling grid in order to refine the vectors that were interpolated in the previous iteration.

The matching error is calculated on blocks of sizes $2X$ and $2Y$, but the best vector is assigned to smaller blocks with dimensions X and Y. This feature is called *block overlapping*, because the larger $2X \cdot 2Y$ block overlaps the final $X \cdot Y$ block in horizontal and vertical direction. It contributes to improve the coherence and reliability of the motion vector field.

Finally, since the calculational effort required for a block matcher is almost linear with the pixel density in a block, we also introduce a *pixel subsampling* factor of four. Hence there are $2X \cdot 2Y/4$ pixels in a large $2X \cdot 2Y$ block where the matching error is calculated, for every iteration. Again, from an iteration to the following, we change also the pixel subsampling grid to spread the number of matching pixels. Fig. 9 shows the calculation and assignment area and the two phases of block and pixel subsampling.

4.4 Final remarks

This new block matching motion estimator can calculate the objects true–

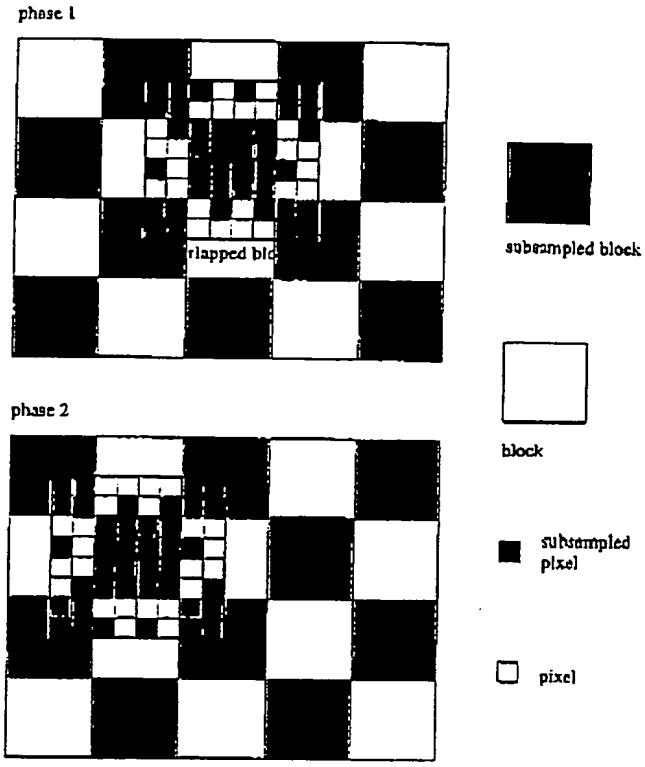

Figure 9: Block overlapping, block subsampling and pixel subsampling.

motion with great accuracy, yielding a very coherent motion vector field, from the spatial and temporal points of view. This means that the VLC differential encoding of macroblock vectors should achieve lower bitrates in comparison with vectors estimated from "classical" full–search block matchers. Furthermore its very low complexity over–compensates the increased global processing complexity, due to the introduction of the MVPF stage in both the encoding and decoding stages.

5 MOTION VECTORS POST–FILTERING

In this Section we will describe the real innovative part of our proposal, the motion vectors post–filtering (MVPF).

In practice, we want to use the overlapped block motion compensation, as it is actually specified in APM of H.263 standard, in both the encoding and decoding terminals, while transmitting and receiving only MB motion vectors (to not increase the bitrate). This means that both terminals have to use the same MVPF, to re–assign the MB vectors to blocks of $8 \cdot 8$ pixels, as performed in the motion estimation part of APM. Fig. 10 shows the temporal prediction block diagram including the MVPF.

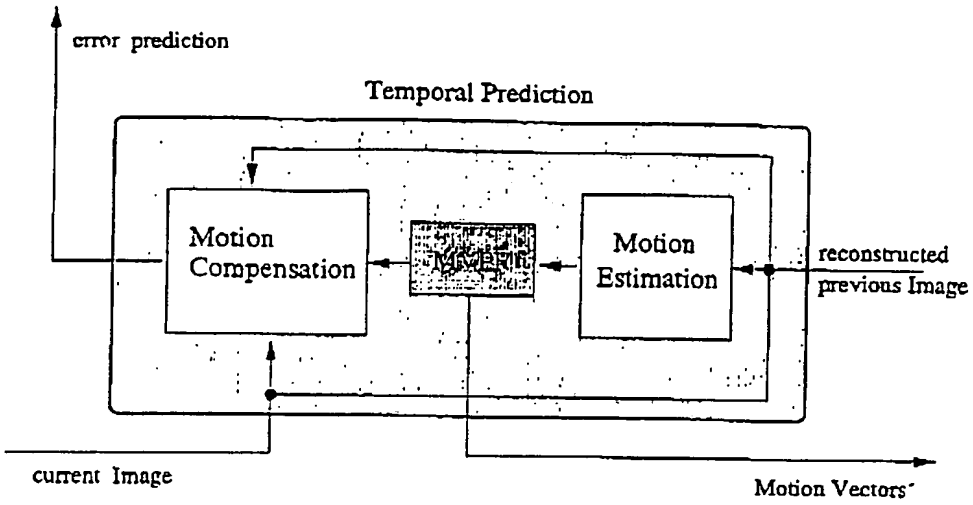

Figure 10: The MVPF in the temporal prediction block diagram.

Even if the MVPF should not be dependent on the estimation strategy, we strongly recommend to use it jointly with the motion estimator described in Section 4, to obtain the best performances. Of course there are several solutions to calculate the block vectors, for example by a weighted averaging of the adjacent macroblock vectors, anyway we will describe in detail only what we consider the best solution, due to the inherent features of our new motion estimator, the *block erosion* MVPF.

5.1 Block erosion

As reported in the previous Sections, in the H.263 standard the motion information is limited to one vector per macroblock of $X \cdot Y = 16 \cdot 16$ pixels, therefore the MVPF performs a *block erosion* to eliminate fixed block boundaries from the vector field, by re–assigning a new vector to a block of sizes $(X/2) \cdot (Y/2) = 8 \cdot 8$.

If $MVc = \vec{d}(\vec{b}_c, t)$ is a macroblock vector centered in $\vec{b}_c$ and its four adjacent macroblock vectors are given by:

$$MVl = \vec{d}(\vec{b}_c - \begin{pmatrix} X \\ 0 \end{pmatrix}, t)$$

$$MVr = \vec{d}(\vec{b}_c - \begin{pmatrix} -X \\ 0 \end{pmatrix}, t)$$

$$MVt = \vec{d}(\vec{b}_c - \begin{pmatrix} 0 \\ Y \end{pmatrix}, t)$$

$$MVb = \vec{d}(\vec{b}_c - \begin{pmatrix} 0 \\ -Y \end{pmatrix}, t)$$

the four $8 \cdot 8$ blocks, numbered as in Fig 2, will be assigned their new vectors according

to the following:

$$MV1{=}median(MVl, MVc, MVa)$$
$$MV2{=}median(MVa, MVc, MVr)$$
$$MV3{=}median(MVl, MVc, MVb)$$
$$MV4{=}median(MVr, MVc, MVb)$$

Figure 11 shows the block erosion of a macroblock vector MVc into four block vectors $MV1, MV2, MV3, MV4$.

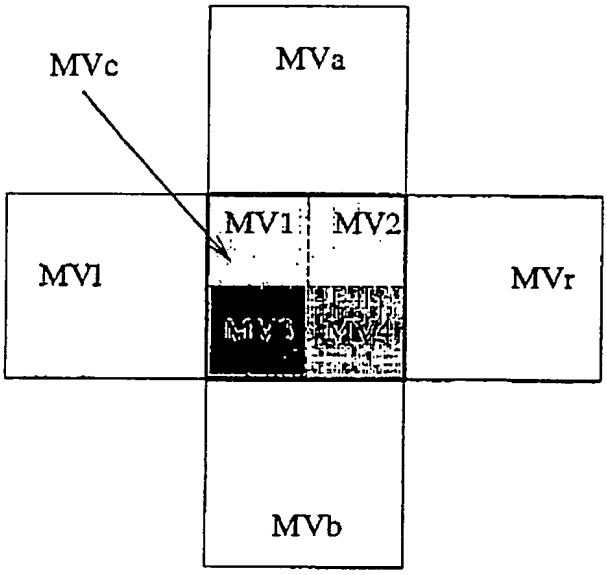

Figure 11: Block erosion: from one vector per macroblock to one vector for every block.

5.2 Standard compatibility

As far as we know, this solution has not been mentioned in the standard, but it is fully H.263 compatible. At the start of the multimedia communication the two terminals exchange data about their processing standard and non–standard capabilities (see [5] for more details). If we assume that, during the communication set–up, both terminals declare this MVPF capability, they will easily interface with each other. Hence, the video encoder will transmit only MB vectors, while the video decoder will post–filter them in order to have a different vector for every block. In the temporal interpolation process both terminals use the overlapped block motion compensation, as it is specified in the H.263 APM. Thanks to this method, we can achieve the same image quality as if the APM was used, but without increasing the bitrate.

If at least one terminal declares to have not this capability, a flag can be forced in the other terminal to switch it off.

6 EVALUATION

To simulate the H.263 encoding process we took the Internet public domain C–language software *tmn-1.6*, written by Telenor R&D [6] during the H.263 Test Model TMN 5 definition phase [9]. Although we briefly present only Tables related to *Teeny*, a girl rotating her head very quickly (see Fig 12), the observations that we will do are generally valid also for other video sequences with CIF, QCIF or SQCIF format.

Figure 12: The *Teeny* test CIF video sequence.

Table 2 shows the behaviour of *Teeny*, encoded with target bitrate of 256 kbps at CIF resolution, with and without APM (and UMV jointly, since H.263 standard suggests their joint action). The mean picture quantizer QP decreases while decreasing the frame rate, indicating that the spatial quality increases necessary at expense of the frame rate [1]. Also the luminance Mean Square Error (MSE) between the original image and its decoded version is clearly decreasing, indicating not only that the spatial detail is increasing, but also that the temporal prediction is still good, due to the robustness of the encoding DPCM/DCT loop.

It is interesting to note that the motion vectors (macroblock information) need from 13 % up to 18 % of the total bitrate in the basic H.263 standard, and 19–25 % in the H.263 with APM and UMV. In this last case the mean quantizer is increased and then the image sharpness is decreased, since the DCT quantization becomes more coarse to "compensate" the extra bit–budget required by the motion vectors block information, while the output buffer controller maintains constant the global bitrate. Finally, with APM and UMV the motion information is increased of about 40 % in comparison with the basic H.263 standard.

[1] A QP closed to 31 means coarse quantization and hence very poor spatial resolution; for $QP \leq 20$ the detail is quite satisfactory.

frame rate	obtained bitrate	motion vectors	mean QP	encoding MSE
12.5 Hz	258.50 kbps	45.66 kbps	17.88	80.84
8.33 Hz	261.12 kbps	34.00 kbps	15.53	70.68

frame rate	obtained bitrate	motion vectors	mean QP	encoding MSE
12.5 Hz	258.6 kbps	64.23 kbps	20.35	79.22
8.33 Hz	261.15 kbps	49.13 kbps	18.51	72.20

Table 2: *Teeny* encoded at a target bitrate of 256 kbps without (top) and with (bottom) APM and UMV.

Thanks to our method, we can use this amount of bits for relaxing the DCT coefficients quantization instead of encoding the motion vectors information related to blocks, so that we achieve higher sharpness pictures than actual H.263 standard encoders with APM, without increasing the bitrates.

On the other hand, if the DCT coefficients quantization is not relaxed, we can encode and transmit "typical H.263 plus APM quality" pictures, while reducing the bitrate because of no block motion information transmission, thus increasing the channel efficiency.

Finally, in our method every block will be assigned its own motion vectors, while in the APM of H.263 standard not all the macroblocks will be processed as four separate blocks. In other words, in APM is always possible that there will remain a consistent number of macroblocks to which a motion vector is assigned, while our method always assigns one proper motion vector to every block.

7 CONCLUSIONS

The invention relates to a low bitrate video coding method fully compatible with H.263 standard and comprising a Motion Vectors Post–Filtering (MVPF) step. This MVPF step assigns a different motion vector to every block composing a macroblock, starting from the original motion vector of the macroblock itself. In this way the temporal prediction is based on $8 \cdot 8$ pixels blocks instead of $16 \cdot 16$ macroblocks, as actually is done when the negotiable option called Advanced Prediction Mode (APM) is used in the H.263 encoder. The video decoding terminal has to use the same MVPF step to produce the related block vectors.

Furthermore, since only macroblock vectors are differential encoded (in a variable length fashion) and transmitted, a considerable bitrate reduction is also achieved, in comparison with APM.

This method is not yet H.263 standardized, so it has to be signalled between the two terminals, via the H.245 protocol. It can be used at CIF, QCIF and SQCIF resolution.

References

[1] ITU–T DRAFT Recommendation H.263, Video coding for low bit rate communication, 2 May 1996.

[2] K. Rijkse, "ITU standardisation of very low bit rate video coding algorithms", Signal Processing: Image Communication 7, 1995, pp 553–565.

[3] ITU–T DRAFT Recommendation H.261, Video codec for audio–visual services at px64 kbits, March 1993.

[4] Telenor Research, "Video Codec Test Model TMN 5", Document ITU-T LBC-95, January 1995.

[5] ITU–T DRAFT Recommendation H.245, Control protocol for multimedia communications, 27 November 1995.

[6] Available on ftp site: *bonde.nta.no* at the directory: *pub/tmn/software.*

[7] G. de Haan, P.W.A.C. Biezen, H. Huijgen, O. A. Ojo, "True motion estimation with 3–D recursive search block matching", *IEEE Trans. Circuits and Systems for Video Technology*, Vol. 3, October 1993, pp. 368–379.

[8] G. de Haan, P.W.A.C. Biezen, "Sub–pixel motion estimation with 3–D recursive search block–matching", *Signal Processing: Image Communication* 6 (1995), pp. 485–498.

[9] Telenor Research, "Video Codec Test Model TMN 5", Document ITU-T LBC-95, January 1995.

[10] P. Lippens, B. De Loore, G. de Haan, P. Eeckhout, H. Huijgen, A. Loning, B. McSweeney, M. Verstraelen, B. Pham, J. Kettenis, "A video signal processor for motion–compensated field–rate upconversion in consumer television", *IEEE Journal of Solid–state Circuits*, Vol. 31, no. 11, November 1996, pp. 1762–1769.

CLAIMS (SECTION 8) :

1. A motion estimation method for a low bitrate video encoder or decoder of signals for which a hierarchical bitstream syntax is defined including macroblock and block levels, said method being based on the use of the so-called advanced prediction mode in terms of motion estimation and compensation, wherein only one motion vector per macroblock is transmitted from the encoding side, while a non-linear operation is used on the decoding side for the motion estimation per block and on the encoding side for the picture prediction.

2. A method according to claim 1, wherein said non-linear operation is a motion vector post-filtering step.

3. A method according to anyone of claims 1 and 2, wherein said step is performed on transmitted vectors corresponding to macroblocks of 16 x 16 picture elements, in order to have a different motion vector for each block of 8 x 8 picture elements.

4. A method according to claim 3, wherein said post-filtering step performs a block erosion in order to eliminate fixed block boundaries.

5. A method according to anyone of claims 1 to 4, wherein the bit-budget saved by encoding and transmitting only macroblock vectors is re-used for a less coarse quantization within the encoder.

6. A motion estimation device for implementing a method according to anyone of claims 1 to 5.

7. An H.263-like video encoder able to make use of the so-called prediction mode in terms of motion estimation and compensation, including in the motion estimation stage of its temporal prediction loop a device according to claim 6.

8. An H.263-like video decoder including in its temporal interpolation stage a device according to claim 6.